

Identificació i aplicació objectes predefinits del llenguatge. Number	1
Exemple	1
Constants	2
Mètodes	2
isFinite()	2
isNaN()	2
Number()	3
parseInt()	3
parseFloat()	3
Operadors aritmètics	3
Exercicis	4

## Identificació i aplicació objectes predefinits del llenguatge. Number

Una variable és de tipus “number” quan emmagatzema un número. Es poden assignar diferents tipus de valors (enter, decimal) i bases (base octal, decimal i hexadecimal) a les variables. Així doncs, les variables de tipus número es poden assignar de diferents maneres:

Amb un número enter        `var = 17;`  
Amb un número decimal     `var=0.17;`  
Amb notació exponencial   `var=17e3;`

### Exemple

```
<!DOCTYPE html>
<html>
  <head>
    <title>Exemple números</title>
  </head>
  <body>
    <script>
      //declarem la variable a
      var a;
      //mostrem el valor de la variable per consola
      console.log(a); // undefined
      //mostrem el tipus de la variable per consola
```

```
        console.log(typeof a); // undefined
        //assignem el número 17
        a = 17;
        console.log(a); // 17
        console.log(typeof a); // number
        a=0.17;
        console.log(a); // 0.17
        console.log(typeof a); // number
        a=17e3;
        console.log(a); //17000 en base 10
    </script>
</body>
</html>
```

## Constants

- MAX\_VALUE: retorna el número més gran disponible a JavaScript
- MIN\_VALUE: retorna el número més petit disponible a JavaScript
- NaN: Representa el valor Not a Number

## Mètodes

Les propietats i mètodes globals de JavaScript es poden utilitzar amb tots els objectes JavaScript.

JavaScript disposa d'una sèrie de funcions predefinides, és a dir, que ja estan integrades al llenguatge. Podem utilitzar aquestes funcions en qualsevol moment sense declarar-les prèviament ni saber quines instruccions executen. Simplement hem de conèixer el nom de la funció i el resultat que obtenim a l'executar-la. Les funcions predefinides més rellevants que podem fer servir amb el tipus number són:

### isFinite()

El mètode isFinite() retorna True si un valor és un nombre legal finit.  
El mètode isFinite() retorna fals si el valor és Infinity, -Infinity o NaN.

### isNaN()

NaN és l'abreviatura de Not a Number.  
El mètode isNaN() retorna cert si un valor és NaN.  
El mètode isNaN() converteix el valor en un nombre abans de provar-lo.  
El mètode isNaN() és diferent de Number.isNaN().

`Number.isNaN()` no converteix el valor abans de provar-lo i retorna `false` si el valor no és del tipus de dades `Number`

## Number()

El mètode `Number()` converteix un valor en un nombre.  
Si el valor no es pot convertir, es retorna `NaN`.

**i** Per als booleans, `Number()` retorna `0` o `1`. Per a les dates, `Number()` retorna mil·lisegons des de l'1 de gener de 1970 a les 00:00:00. Per a les cadenes, `Number()` retorna un número o `NaN`.

## parseInt()

El mètode `parseInt` analitza un valor com una cadena i retorna el primer nombre enter.

**i** Si el primer caràcter no es pot convertir, es retorna `NaN`. Els espais inicials i finals s'ignoren. Només es retorna el primer nombre enter trobat.

## parseFloat()

El mètode `parseFloat()` analitza un valor com una cadena i retorna el primer número.

## Operadors aritmètics

Recordem que els operadors aritmètics s'utilitzen amb variables de tipus `Number` o `String`. Es poden diferenciar les següents operacions:

- Suma `a=b+c;`
- Resta `a=b-c;`
- Multiplicació `a=b*c;`
- Exponent `a=b**c;` `//a=b^c`
- Divisió `a=b/c;`
- Mòdul `a=b%c`
- Increment `a++;`
- Decrement `a--;`

Alternativament, alguns operadors aritmètics poden incloure l'operador d'assignació:

- Suma `a+=b;` `// a=a+b;`
- Resta `a-=b;` `// a=a-b;`
- Multiplicació `a*=b;` `// a=a*b;`
- Divisió `a/=b;` `// a=a/b;`
- Mòdul `a%=b;` `// a=a%b;`

## Exercicis

1. Fes un programa on es forci el resultat NaN i es mostri per pantalla.
2. Fes un programa per convertir un String nombre a nombre.
3. Fes un programa per validar que no es un número.